In the Specification

Please amend the specification of this application as follows:

Rewrite the paragraph at page 8, line 28 to page 9, line 6 as follows:

-- The portable computer 310 superlink function 360 is adapted to handle a plurality of different types of data exchange, e.g., LPC, Ethernet, USB, legacy (mouse, printer, audio, etc.), IRDA, AC-97, ~~I2S,~~ I^2S, SMBus, GPIO's, Docking Sideband, hot dock/undock. A PCI link enable and status function is included. Advantageously, the LPC I/O bridge device 360 is adapted to serialize data transmission of a plurality of protocols, such as legacy I/O protocols, keyboard controller, mouse controller, PS2, AT keyboard, parallel, serial, GPIO's, AC-link, SMBus, SPDIF/Audio, and LPC bus, as examples.--

Rewrite the paragraph at page 10, lines 3 to 15 as follows:

-- Figure 6 illustrates a more detailed block diagram of an embodiment of the present LPC I/O bridge device 360. The LPC-based super I/O function 340 includes a serial interface 358 and is coupled to I/O bus 356. I/O bus 356 is coupled to a plurality of GPIO ports 362, e.g., 32, with event notification, fan speed and control 366, e.g., for two fans, a watchdog timer 364, SMBus controller 374, system wake-up control (not shown), parallel port ~~374~~ 370 such as IEEE 1284, a serial port 368, a floppy disk controller 378, serial IRQ interface 358, docking control 380. LPC I/O bridge device also includes an AC97 interface 390 and a USB hub 344. A packetizer/depacketizer 382 adapted to packetize and depacketize serialized data is coupled to LPC controller 340 and AC97 390, as shown. Preferably, an advanced CMOS process is used to achieve low system-power consumption. The device 360 may also include configuration and control registers 376, FIFO 384, serdes

386, HS I/O 388, strap configuration 372, and a serial EPROM interface 392, as shown.--

Rewrite the paragraph at page 10, lines 16 to 29 as follows:

-- The LPC-based super I/O function or controller 340 may include a plurality of features, including ACPI 348 with supported wake events for ring indicate, fan tachometer, GPIO, dock and undock events and including automatic power down for serial port, parallel port and FDC. ACPI 348 is adapted to support SMI (legacy). Serial IRQ 358 supports active and quiet modes, is fixed to 21 frames, and has no external parallel IRQ. SMBus 374 is preferably slave function only and preferably has access to global register set. Fan speed and control monitor 366 may support e.g., two fan control and monitor units, with the fan speed control being a programmable PWM generator. A floppy drive controller 378 is included, as well as 32 GPIO ports 362, each port including event detection and notification, with 8 ports powered by V_{SB} providing event notification and wake-up. Watchdog timer 364 looks for four maskable system events. An LPC Interface 354 and AC'97 Interface 390 with support for two codecs, for example, are included, as well as serial ~~EEPROM~~ EPROM Interface 392 and a 4 port USB Hub 344 are also included.--

Rewrite the paragraph at page 11, lines 1 to 7 as follows:

-- The LPC I/O bridge device 360 comprises a plurality of logical devices, a host interface and a central set of configuration registers, each coupled to and built around an internal bus architecture. The LPC Controller 340 preferably comprises a logic chip and functions to serve as a bridge between the external LPC ~~352~~ interface 354 and the internal bus 356. The LPC controller 340 preferably is adapted to support 8 bit I/O read,

8 bit I/O write and 8 bit direct memory access (DMA) transactions on the LPC bus, as examples.--

Rewrite the paragraph at page 11, lines 8 to 22 as follows:

-- General preferences regarding aspects of embodiments of the present invention will next be described, with reference to Figures 4B, Figure 5 and Figure 6. The LPC I/O bridge device 360 is preferably implemented in a portable computer 310 having a plurality of legacy ports 329 coupled to an I/O bus 356 and a docking connector 330. The device 360 includes an LPC controller 340 coupled to the I/O bus 356 and docking connector 330, where the LPC controller 340 is adapted to detect whether the portable computer 310 is coupled to a docking station 312 via the docking connector 330 and route data transmissions from the I/O bus 356 to the legacy ports 329, docking connector 330, or both. Serialization logic 346 is coupled to the LPC controller 340, wherein the serialization logic 346 is adapted to serialize the data transmissions routed to the docking connector 330. The LPC controller 340 is coupled to ~~an~~ LPC interface 354, wherein routing decisions are based on information received on the LPC interface 354, such as, the type of data and address information received from the BIOS or operating system. Preferably, at least the LPC controller 340 and serialization logic 346 reside on a single integrated circuit.--

Rewrite the paragraph at page 11, lines 23 to page 12, line 2 as follows:

-- The portable computer 310 may comprise a variety of portable computers such as a notebook or laptop computer, PDA, or wearable computer. The docking station 312 adapted to receive the serialized data transmissions from the LPC I/O bridge device 360 ~~of Claim 1~~ through the portable computer docking connector 330.

Preferably, the docking station 312 includes a single integrated circuit 362 adapted to control the receipt of the serialized data transmission. The docking station may comprise a docking station such as the one shown in Figure 1, a port replicator or expansion chassis, as examples.--

Rewrite the paragraph at page 21, lines 23 to 27 as follows:

-- The SMBus allows easy interfacing to a wide range of low-cost memories and I/O devices, including ~~EEPROMs.~~ EPROMs, SRAMs, timers, ADC, DAC, clock chips and peripheral devices. With System Management Bus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status.--

Rewrite the paragraph at page 24, lines 3 to 11 as follows:

-- A high-to-low transition on the SMBCLK will cause all devices involved to start counting off their low period and start driving SMBCLK low ~~id~~ if the device is a master. As soon as the device finished counting its low period it will release the SMBCLK line. However, the low-to-high transition of the clock may not change state if another master with a longer low period keeps the SMBCLK line low. In this situation the master that released the SMBCLK line will enter the WMBCLK high wait period. When all devices have counted off their low period the SMBCLK line will be released and go high. The first device that completes the high period count will pull the SMBCLK line low and the cycle will start again.--

Rewrite the paragraph at page 25, lines 23 to 27 as follows:

-- The PWM Counter preferably comprises an 8-bit free-running counter that is cyclical. The FCDCR register contains the duty cycle so that when the counter exceeds the value the comparator

will go low. The clock divider is controlled by the FCPSR register, supports a division factor of 1 to 124. The FCPSR register also controls the selection of the clock input.--

Rewrite the paragraph at page 28, lines 12 to 16 as follows:

--When used in a docked environment, superlink 360 provides for total of 64 GPIO pins, arranged in 8 8-bit ports. Some ports are not accessible unless docked. Several of the GPIO port pins are multi-function pins and can be configured to support specific use functions. Among the functions are Fan Control and Monitor functions and Watchdog timer user defined inputs. ~~Some GPIO ports--~~

Rewrite the paragraph at page 31, lines 9 to 11 as follows:

-- The ACE contains a programmable baud generator that takes a clock input and divides it by a divisor in the range between 1 and ~~(2^16 -1)~~ $(2^{16} -1)$. The output frequency of the baud generator is 16x the baud rate. The formula for the divisor is:--

Rewrite the paragraph at page 32, lines 3 to 9 as follows:

-- Figure 13 illustrates a block diagram of a preferred embodiment for the high speed serial interface 388 of Figure 6. The serial interface block preferably comprises an ~~8b10b~~ 8B/10B encoder/decoder and the high speed serial interface. The high speed serial interface preferably comprises two signal pairs operating at a standard line speed, e.g., 1.25 Gbps, 1.6 Gbps, or 2.5 Gbps, as examples. The signal pairs preferably operate in dual simplex fashion. The data is encoded using the ~~8b10b~~ 8B/10B encoding that is common to Gigabit Ethernet transmission.--

Rewrite the paragraph at page 32, lines 26 to page 33, line 3 as follows:

-- The data transmission latency is defined as the delay from the initial 8-bit word load to the serial transmission of bit 0. The transmit latency is preferably specified to be 10 bit times. Note that this latency preferably only applies to the portion of the circuitry starting from the ~~8b10b~~ 8B/10B encoder to the transmission of bit 0 on the serial interface. Additional latencies exist within the system which include packetization, FIFO depth and arbitration times. This will have some variance to it, which makes determining an exact latency difficult.--

Rewrite the paragraph at page 33, lines 26 to page 34, line 3 as follows:

--A preferred embodiment for the ~~8b10b~~ 8B/10B encoder/decoder will next be described. True serial interfaces require a method of encoding to insure minimum transition density so that the receiving PLL has a minimal number of transitions in which to stay locked on. The encoding scheme maintains the signal DC balance by keeping the number of ones and zero's the same. This provides good transition density for clock recovery and improves error checking. The serial interface uses the ~~8b10b~~ 8B/10B encoding algorithm that is used by Fibre channel and Gigabit Ethernet.--

Rewrite the paragraph at page 34, lines 4 to 14 as follows:

--The PCS layer maps the GMII signals into ten-bit code groups and vice versa, using an ~~8b10b~~ 8B/10B block coding scheme. The PCS uses the transmission code to improve the transmission characteristics of information to be transferred across the link. The encodings defined by the transmission code ensure that sufficient transitions are present in the PHY bit stream to make clock recovery possible in the receiver. Such encoding also greatly increases the likelihood of detecting any single or multiple bit errors that may occur during transmission and reception of

information. The ~~8b10b~~ 8B/10B transmission code specified for use has a high transition density, is run length limited and is DC-balanced. The transition density of the ~~8b10b~~ 8B/10B symbols range from 3 to 8 transitions per symbol. The definition of the ~~8b10b~~ 8B/10B transmission code is specified in IEEE 802.3z and ANSI X3.230-1994 (FC-PH), clause 11. --

Rewrite the paragraph at page 34 lines 15 to 22 as follows:

--~~8b10b~~ 8B/10B transmission code uses letter notation describing the bits of an unencoded information octet. The bit notation of A,B,C,D,E,F,G,H for an unencoded information octet is used in the description of the ~~8b10b~~ 8B/10B transmission code-groups, where A is the LSB. Each valid code group has been given a name using the following convention: /Dx.y/ for the 256 valid data code-groups and /Kx.y/ for the special control code-groups, where y is the decimal value of bits EDCBA and x is the decimal value of bits HGF (notated as K<HGF.EDCBA>). Thus, an octet value of FE representing a code-group value of K30.7 would be represented in bit notation as 111 11110. --

Rewrite the paragraph at page 35, lines 8 to 21 as follows:

-- The 8B/10B decoder converts 10 bit encoded data back into 8 bits. A comma detect circuit is included that is designed to provide for byte synchronization to an ~~8b/10b~~ 8B/10B transmission code. When parallel data is clocked into a parallel to serial converter, the byte boundary that was associated with the parallel data is now lost in the serialization of the data. When the serial data is received and converted to parallel format again a way is needed to be able to recognize the byte boundary again. Generally, this is accomplished using a synchronization pattern. This is generally unique a pattern of 1's and 0's that either cannot occur as part of valid data or is a pattern that repeats at defined

intervals. ~~8b/10b~~ 8B/10B encoding contains a character called the comma (b'0011111' or b'1100000') which is used by the comma detect circuit to align the received serial data back to its original byte boundary. The decoder detects the K28.5 comma, generating a synchronization signal aligning the data to their 10 bit boundaries for decoding. It then converts the data back into 8 bit data, removing the control words.--

Rewrite the paragraph at page 36, lines 10 to 27 as follows:

--Figure ~~24~~ 14 illustrates a preferred embodiment of a synchronization state diagram for the high speed serial interface synchronization. The serial interface includes a synchronization state machine which is responsible for handling link initialization and synchronization. Upon power up or reset, the state machine enters the acquisition (ACQ) state and searches for IDLE. Upon receiving 3 consecutive IDLE's or a carrier extend, the state machine will enter the synchronization (SYNC) state. If the state machine receives valid data or an error propagation during the acquisition process, it will immediately transition to the SYNC state. Loss of synchronization occurs whenever four consecutive invalid transmissions have been detected or when 4 invalid transmissions occur prior to receiving 4 consecutive valid data groups or IDLE's. A single invalid transmission received while in the SYNC state will cause the state machine to transition to the loss of synchronization (LOS) state. Receiving three additional invalid transmissions before 4 consecutive valid transmissions occur while in LOS will force the state machine back to the acquisition state. If four consecutive valid transmissions occur then the state machine will transition to the SYNC state. Embodiments of the invention include high speed serial inteface configuration registers, high speed serial configuration registers,

high speed serial TXMODE Configuration Register, and high speed serial status registers, as examples of registers utilized.--

Rewrite the paragraph at page 50, lines 15 to 22 as follows:

-- Low Power Mode and Warm Reset will next be described. If the primary superlink is operating as the primary codec, the primary superlink device will monitor the SDATA_OUT pin and shut down the BIT_CLK when a write occurs to bit PR4 of the Powerdown Control/Status Register (Index 26). ~~superlink~~ Superlink will pass a warm reset across the serial link when acting as either a primary or a secondary codec. When operating as a primary codec, superlink will also restart the BIT_CLK when a warm reset is received. A warm reset comprises SYNC going high for a minimum of 1us in the absence of BIT_CLK.--

Rewrite the paragraph at page 55, line 22 to page 56, line 6 as follows:

-- A preferred embodiment of the serial ~~EEPROM~~ EPROM interface (integrated with USB Hub) 392 of Figure 6 will next be described. The superlink USB Hub has an on-chip end-device (compliant with USB device specification, please refer to Chapter 9, USB Specification Version 1.1) for storing some information, which may be read and written by the host through the USB Hub upstream port. To USB Hub, this end-device is a permanently attached device (which is beyond those 4 downstream ports mentioned in section 14, USB Hub), which has serial interface with outside ~~EEPROM~~ EPROM. It contains 1K (or 2K) memory space. After power-on reset, the serial ~~EEPROM~~ EPROM interface performs a one-time access read operation from the outside ~~EEPROM~~ EPROM. All the data read from the ~~EEPROM~~ EPROM will be stored inside the on-chip memory space. Then the host can access all the information through the USB Hub port. This serial ~~EEPROM~~ EPROM interface provides communication between the superlink and

the ~~EEPROM~~ EPROM, which is an industry standard 2 wire serial bus protocol --- CLOCK and DATA. The superlink can set the ~~EEPROM~~ EPROM output enable to '1', which can be USB power on reset, then access those bits. --

Rewrite the paragraph at page 56, lines 7 to 19 as follows:

--To interface to the outside ~~EEPROM~~ EPROM, after system power-on reset, initially, the DATA signal will be driven by the serial ~~EEPROM~~ EPROM interface to send a start bit (1), which is followed by the read instruction (10) and the starting-word address (00000). Once the read instruction is received, the instruction and address are decoded by the ~~EEPROM~~ EPROM, which then sends the data to the output shift register. At this point, the interface stops driving the DATA signal and the ~~EEPROM~~ EPROM starts driving. A dummy (0) bit is then output and all the data in the ~~EEPROM~~ EPROM are output with the most significant bit (MSB) first. The output data changes are triggered by the rising edge of the clock provided by the serial ~~EEPROM~~ EPROM interface on the CLOCK signal. Any ~~EEPROM~~ EPROM used must have the automatic internal address advance function. After finishing reading all the data from the ~~EEPROM~~ EPROM, the interface may put the ~~EEPROM~~ EPROM interface into a high-impedance condition (pulled down internally) to allow other logic to share the ~~EEPROM~~ EPROM.--